SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
Masters of Computer Applications

MCA Semester I

Subject Name: Data Structures & Analysis of Algorithms **Subject Code:** MTCA13101

Type of course: Professional Core Course

Prerequisite (if any):
- Programming Language with C
- Mathematical Fundamentals

List of Courses where this course will be prerequisite:
- Basic Mathematics
- Object Oriented Programming
- Database management Systems
- Computer Networks
- Machine Learning

Rationale: Data structure and algorithms help in understanding the nature of the problem at a deeper level. This helps the graduate to write efficient code while developing solutions to the problems in computer science.

Teaching and Examination Scheme:

TEACHING SCHEME				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
3	0	4	5	60	25	15	60	40	200

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests **CA2:** Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems **TEE:** Term End Examination **TEP:** Term End Practical Exam (Performance and viva on practical skills learned in course) **CA3:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. No.	Content	Teaching Hrs.	Module Weightage
1	**Introduction to Data Structures** Introduction; Types of Data Structures, Storage Representation of Strings, Asymptotic Theta Notation	**03**	**10%**
2	**Non-Primitive Linear Data Structures:** Arrays, Storage Structure for Arrays Stack : Operations of Stack, Applications of Stacks : Function Call, Recursion, Queue: Types of Queues, operations and its applications Linked List : Types of Linked List, operations and its applications Applications of Linked List : Addition and Multiplication of Polynomial in one and two variables	**14**	**30%**
3	**Non-Primitive Non-Linear Data Structures:** **Graphs**: Introduction, Definition, Directed/Undirected Graphs, Weighted/Unweighted Graphs Path, Paths of different lengths, Cycle, Cylic Graphs, Acylic Graphs, Spanning Trees, Shortest Path. **Trees:** Introduction, Definition, Basic Tree Concepts, Storage Representation of Binary Trees, Tree Traversal, Height Balanced Tree, 2-3 Trees, Weight Balanced Tree	**10**	**20%**
4	**Algorithm Design and Techniques:** 1. Divide and Conquer Method: Binary Search and Merge Sort 2. Greedy Methods: Knapsack Problem 3. Dynamic Programming: Travelling Salesman Problem 4. Backtracking, Branch and Bound: BFS, DFS	**11**	**25%**
5	**Sorting Techniques:** Selection Sort, Bubble Sort, Merge Sort, Heap Sort, Quick Sort, Radix Sort	**07**	**15%**

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	20	15	15	15	15

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

Sr. no.	Title of book /article	Author(s)	Publisher and details like ISBN	Year of public ation	Publication Edition
1	An Introduction to Data Structures with Applications	Jean-Paul Tremblay, Paul G. Sorenson	Tata McGraw-Hill ISBN: 9780070651500	2007	2nd Edition
2	Data Structures and Algorithm Analysis in C	Mark Allen Weiss	Pearson Education ISBN: 8177583581	2002	2nd Edition
3	Design and Analysis of Algorithms	Parag H Dave, Himanshu B Dave	Pearson Education ISBN: 9332585482	2017	3rd Edition
4	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein	The MIT Press, ISBN: 978-0262033848	2009	3rd Edition

Course Outcomes:

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % weightage
CO-1	Ability to analyze algorithms and algorithm correctness based on the running times	10%

CO-2	Ability to describe stack, queue and linked list operations	30%
CO-3	Ability to perform sorting and searching operations	15%
CO-4	Analyze the sorting techniques	25%
CO-5	Understand the fundamentals of Trees and Graphs	20%

Mapping with POs:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
CO-1	3	3	0	0	0	0	0	0	0	0	0	3			
CO-2	3	3	0	2	0	0	0	0	0	0	0	3			
CO-3	3	3	0	2	0	0	0	0	2	0	0	3			
CO-4	3	3	0	1	2	0	0	0	2	0	0	3			
CO-5	3	3	0	1	0	0	0	0	1	0	0	3			
Rationale*															

Rationale*: Explaining why it is matching this particular program outcome

List of Open learning website:

- https://onlinecourses.nptel.ac.in/noc19_cs47/preview
  (NPTEL Course on Design and Analysis of Algorithm. Covers Unit 1)
- https://www.tutorialspoint.com/data_structures_algorithms/index.htm
- https://www.geeksforgeeks.org/data-structures/
- https://www.javatpoint.com/data-structure-tutorial

List of Open Source Software:

- Inbuilt compiler GCC available on UBUNTU

FOR LAB SESSIONS:

List of Experiments:

Sr. No.	Problem Statements
1	Write a program to implement Stack Operations.
2	Write a program to convert an infix arithmetic expression into postfix notation.
3	Write a program to evaluate a postfix expression.
4	Write a program to perform the INSERT and DELETE operations on a simple queue using array.
5	Write a program to perform the INSERT and DELETE operations on a simple queue using linked list.
6	Write a program to perform the INSERT and DELETE operations on a circular queue.
7	Write a Program to implement Double ended queue
8	Write program perform the following operations on a singly linked list. ● Insert an element ● Delete an element ● Find the sum of elements of the list ● Count number of the nodes in the linked list ● Search a given elements in the linked list. ● Reverse the linked list. ● Make a copy of the given linked list ● Concatenate two linked list ● Merge two linked list
9	Write program to sort a given list using ● Bubble sort ● Selection sort ● Insertion sort ● Quick sort ● Merge sort ● Radix sort ●
10	Write program to search an element in a given list using ● Linear search ● Binary search

Major Equipment Needed: NA